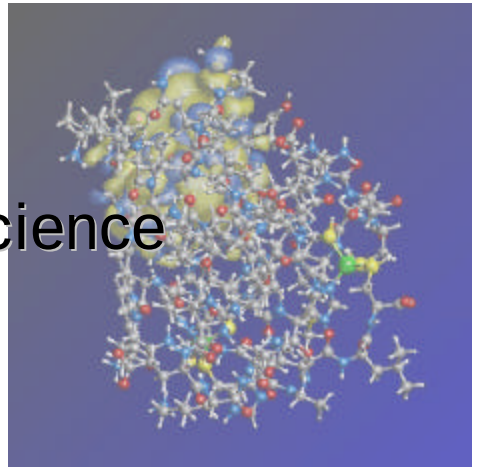
The Grid Datafarm Architecture for Data Intensive Computing

http://datafarm.apgrid.org/

Osamu Tatebe, Satoshi Sekiguchi
National Institute of Advanced Industrial Science and Technology (AIST)
Youhei Morita, Hiroyuki Satoh, Yoshiyuki Watase (KEK)
Satoshi Matsuoka (Tokyo Inst. Tech (TITECH) CC)
Noriyuki Soda (SRA)

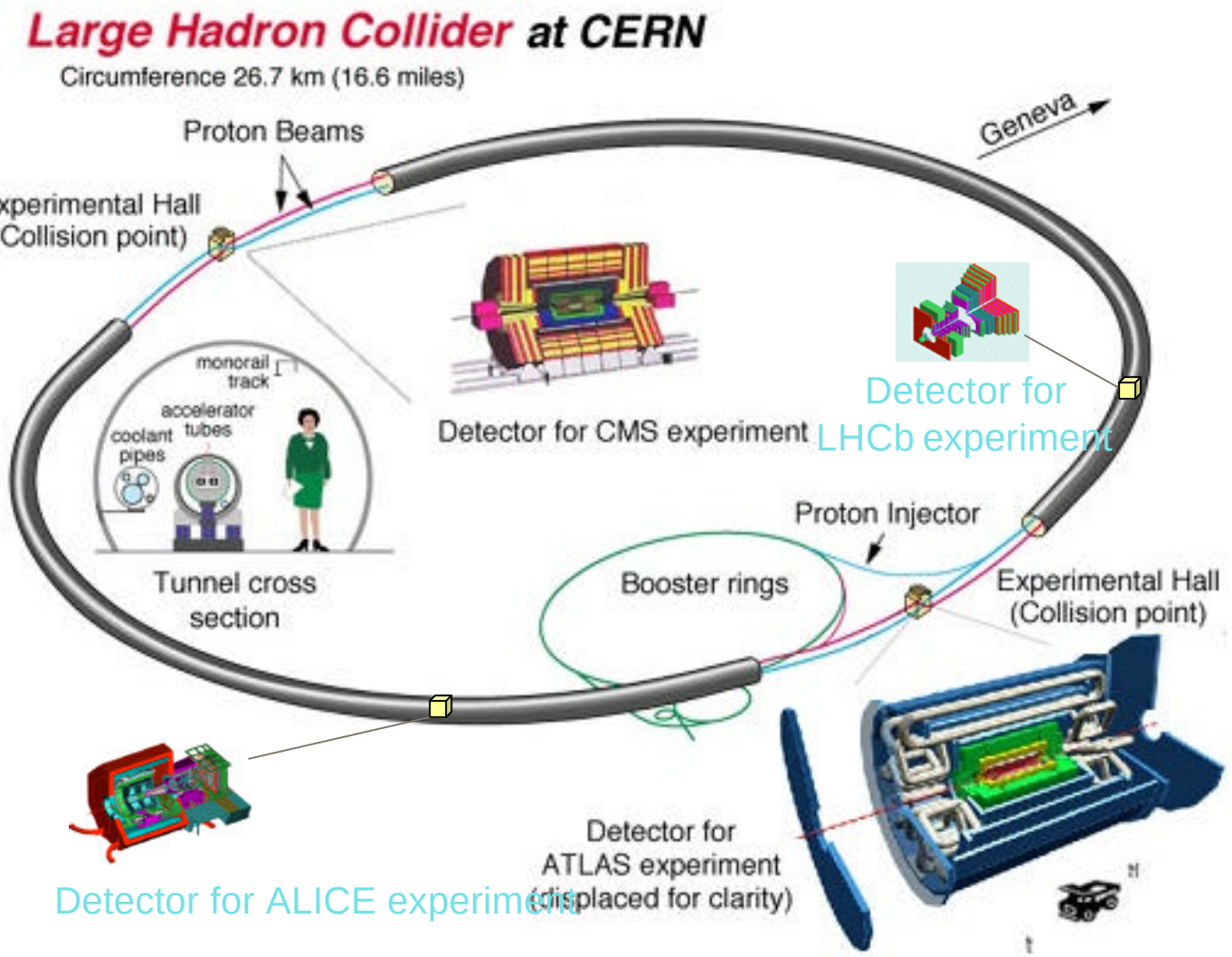
Petascale Data Intensive Computing / Large-scale Data Analysis

- Data intensive computing, large-scale data analysis, data mining
  - High Energy Physics
  - Astronomical Observation, Earth Science
  - Bioinformatics...
  - Good support still needed



- Large-scale database search, data mining
  - E-Government, E-Commerce, Data warehouse
  - Search Engines
  - Other Commercial Stuff

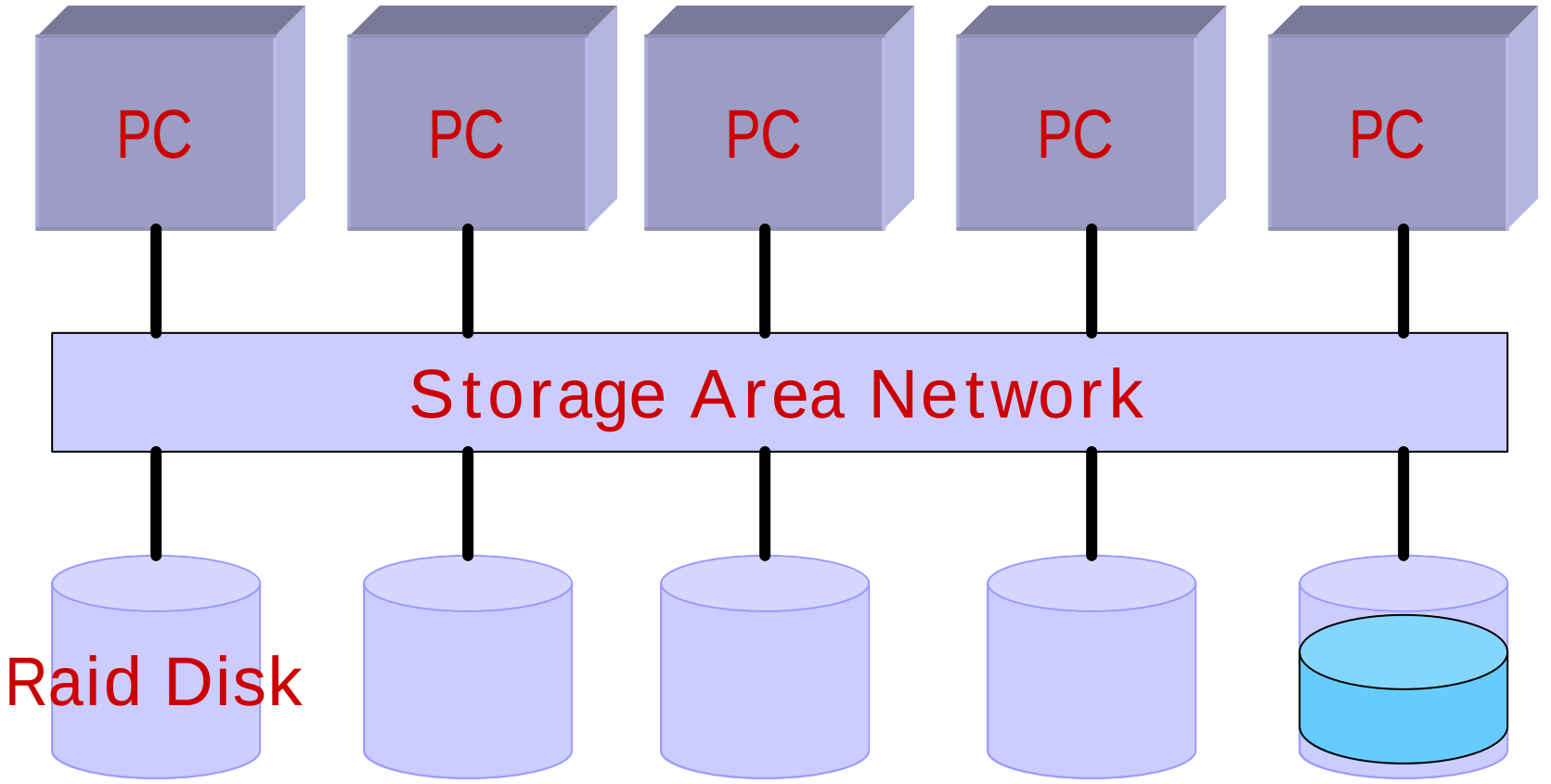
Example: Large Hadron Collider Accelerator at CERN



World-wide Peta/Exascale Data Intensive Computing Requirements

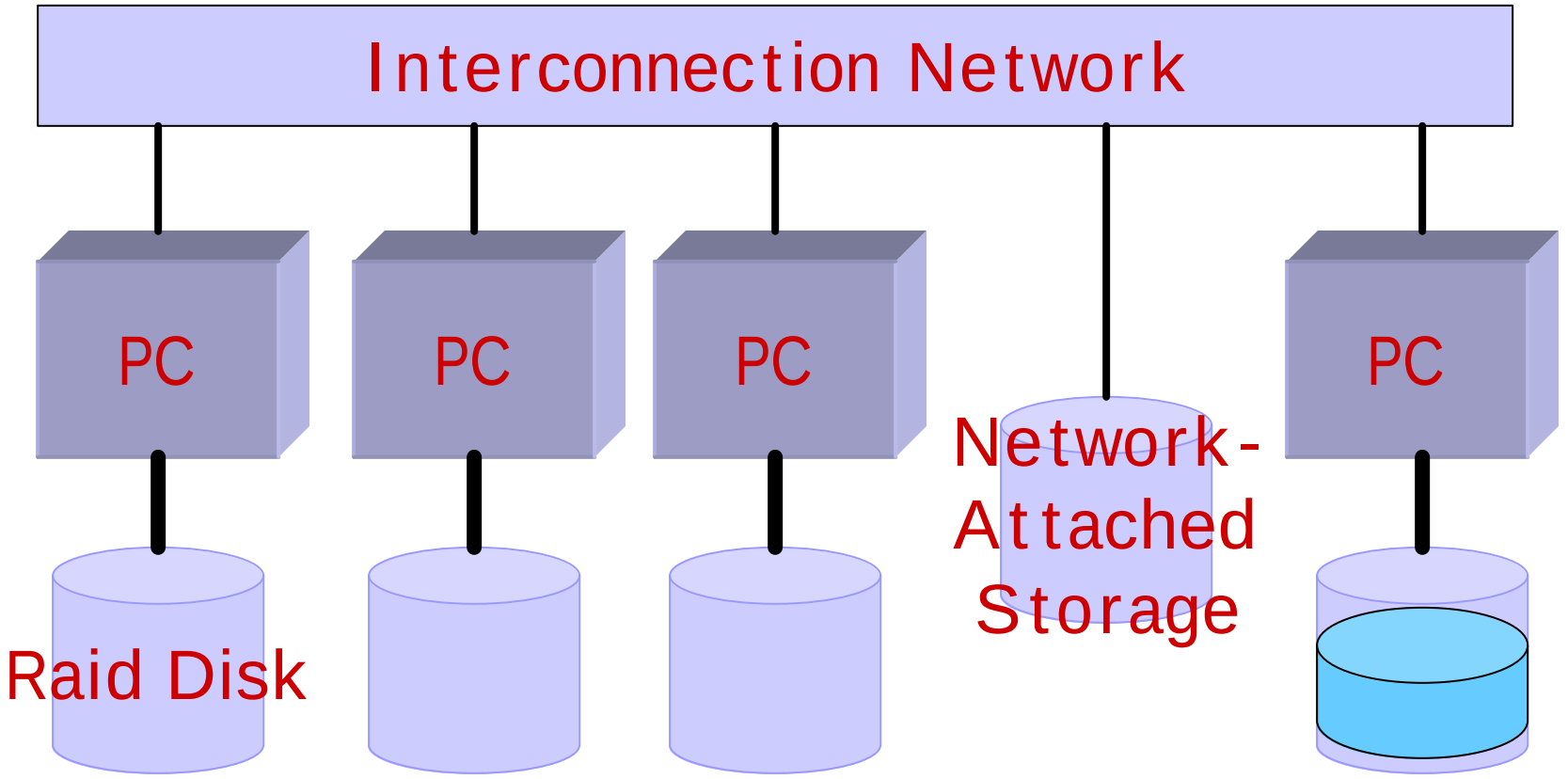
- Peta/Exabyte scale files
- Scalable parallel I/O throughput
  - > 100GB/s, hopefully > 1TB/s
- Scalable computational power
  - > 1TFLOPS, hopefully > 10TFLOPS
- World-wide group-oriented authentication and access control
- Resource Management and Scheduling
- Data / program sharing and efficient access
- System monitoring and administration
- Fault Tolerance / Dynamic re-configuration
- Global Computing Environment

Approach 1 : Storage Area Network (SAN)



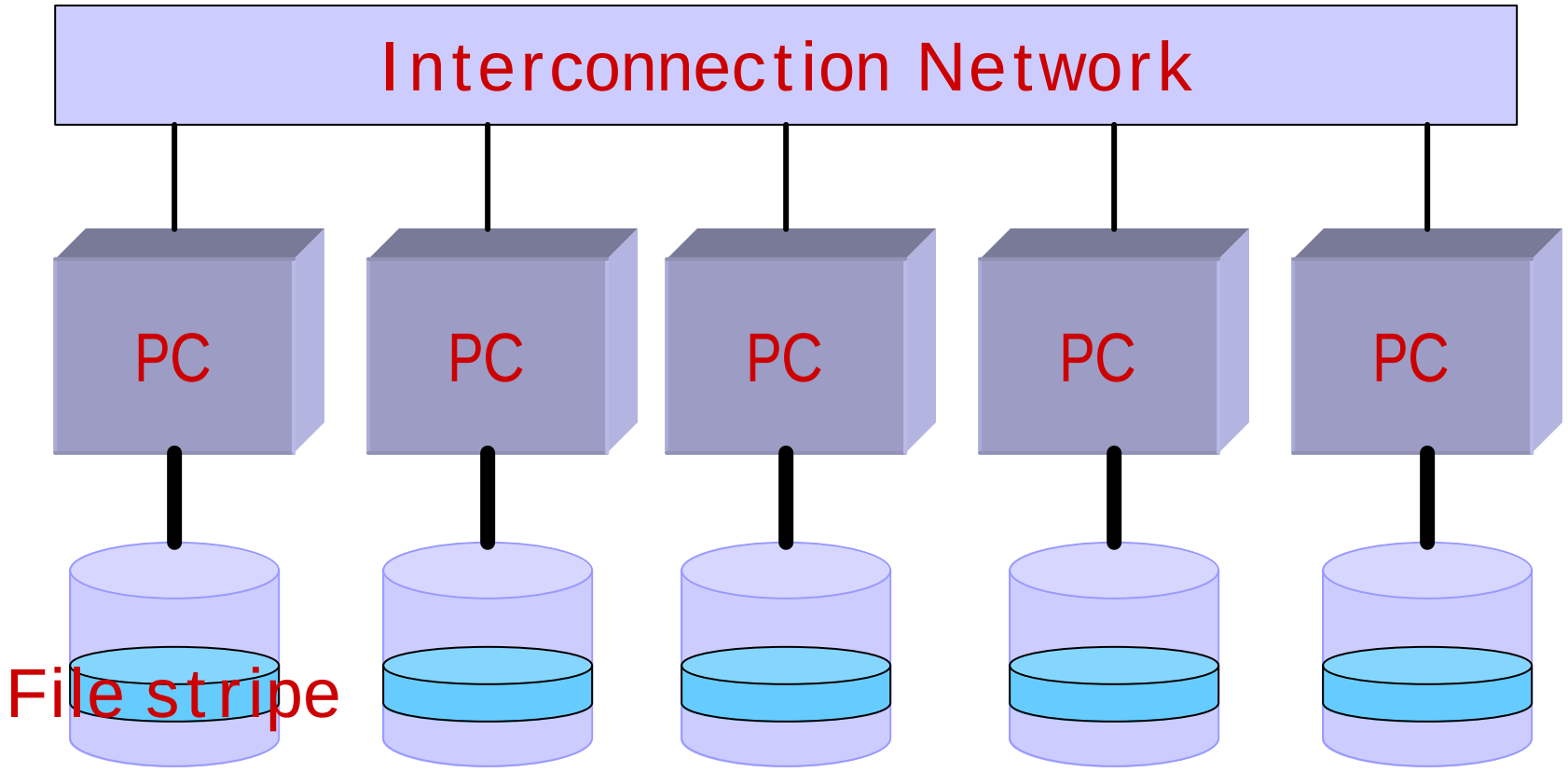
Scalability problems
Grid computing applicability?

Approach 2 : Distributed Filesystem – NFS, NFSv3, AFS, …



Non-single system image
Scalability problems

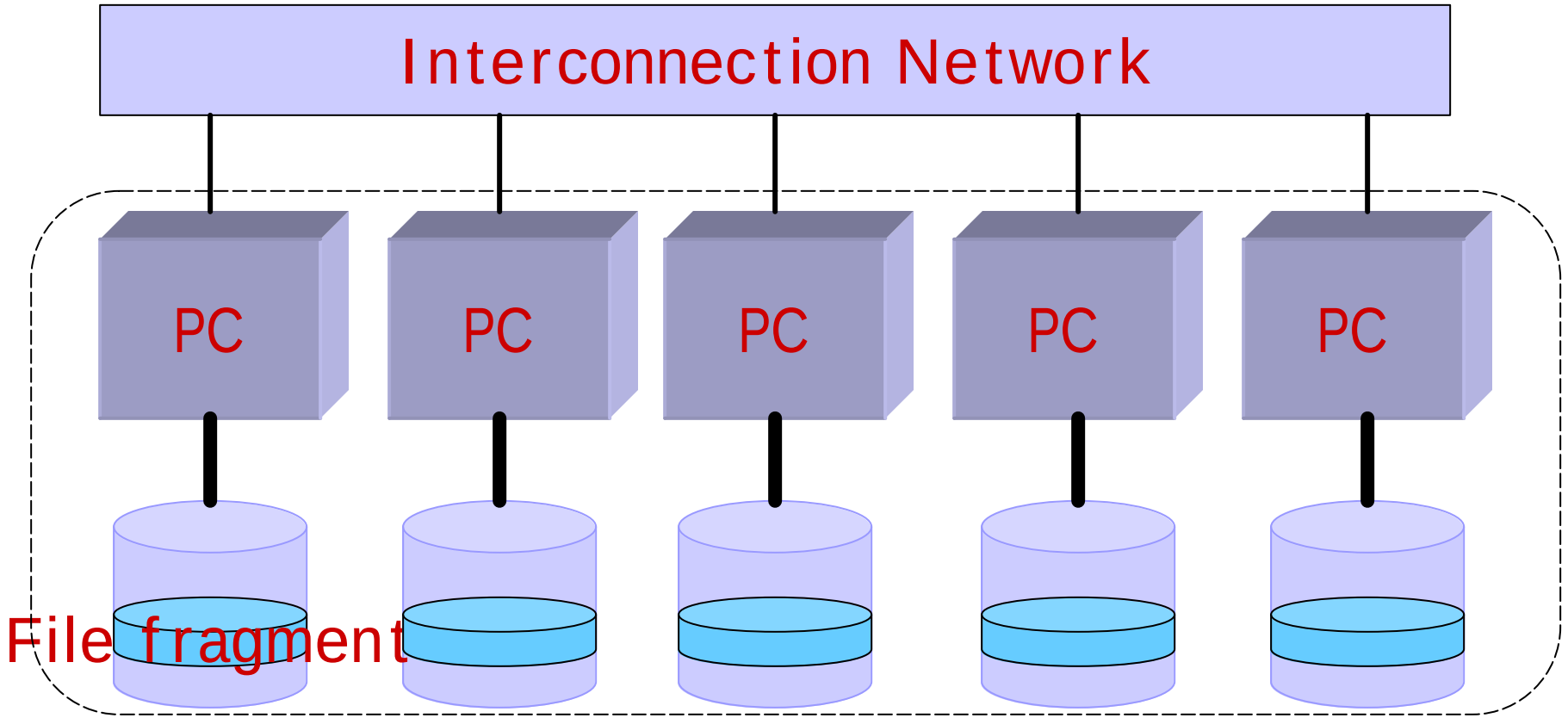
Approach 3: Striping Filesystem



Single system image & Parallel I/O

Tough scheduling problem to maximize I/O bandwidth

Our approach : Distributed Data Parallel Filesystem

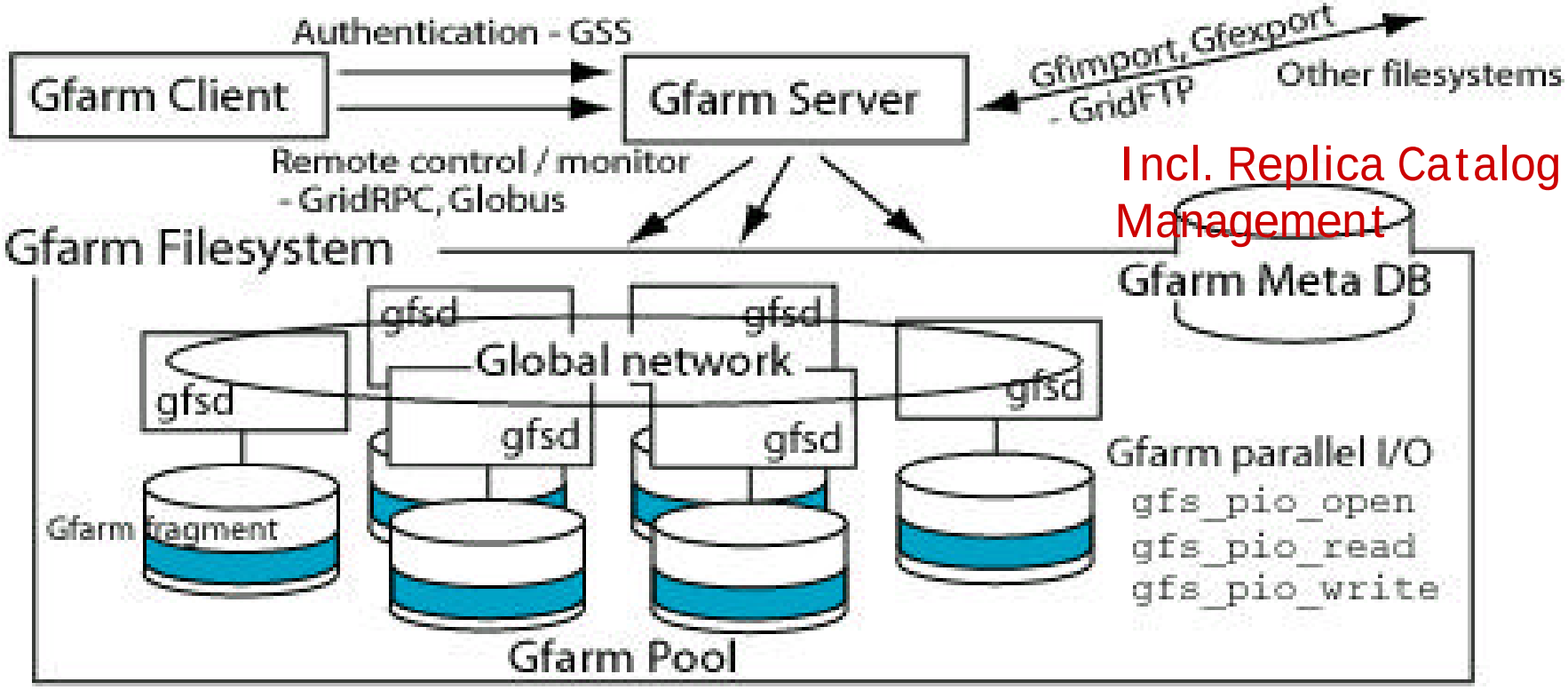


Single system image & Parallel I/O
Exploit Locality of Local I/O channels

Our approach (2) : Distributed Parallel Filesystem for Grid of Clusters

- Tune for Data Intensive Parallel Computing
  - Exploit Disk access locality and bandwidth
  - Local disks utilized for scalable I/O throughput
- Extension of Striping filesystem
- Not only file striping but also Disk-owner computing (caveat: not strictly enforced, for load balancing and FT)
- MPI-IO insufficient especially for irregular and dynamically distributed data
- Grid-aware parallel I/O library for Disk-owner computing and single system image

Design of the Grid Data Farm (1) --- Overview



Peta-to-Exascale Global Filesystem

Parallel I/O and parallel processing

Based on Grid technology & PC Cluster tech.

Design of Grid Data Farm (2) --- Design Principles

- Parallel Gfarm filesystem for on-line Petascale Massively parallel data processing
  - >600Mbps incoming stream catalogued and stored
  - Assume Petascale unified CPU/storage cluster architecture (no SAN) for Massive Parallelism
  - Petascale online storage on distributed files on disk
  - Metadata Management using Databases
- Parallel I/O and Parallel processing
  - Coarse Grained, Task Parallel, Data Intensive Jobs
    - Multiple massive runs, user-specific analysis components
  - Gfarm Toolkit/Shell/Portal hides the Grid complexity
- Base on Grid tech. and Massive PC cluster tech..
  - Ninf(GridRPC), GridFTP, Globus, NWS, etc.

Design of Grid Data Farm (3) --- Design Principles (cont'd)

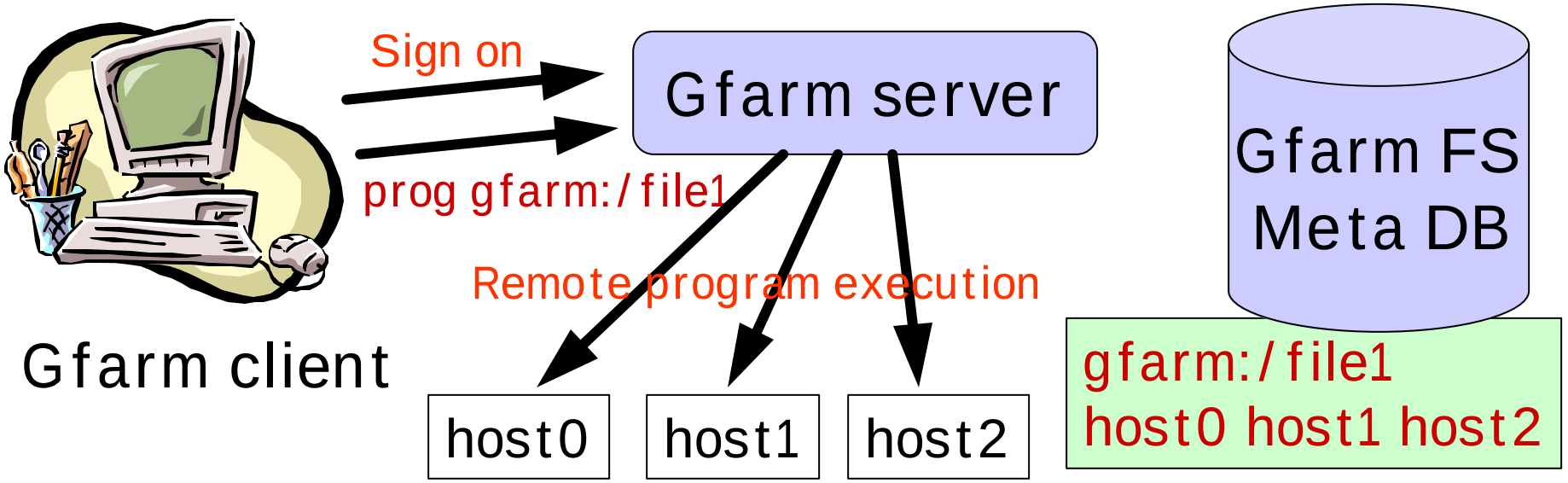
- Fault tolerance
  - Node and disk failures not exceptional cases but common
    - Upstream data too large to be replicated often
  - Recovery via combined strategies, incl. short-range replicas and selective recomputation
    - All data have history and replica Metadata
    - Computation Reproducible
    - Synergy of Replication and Load Balancing
- Dynamic, automated load-balancing
  - via dynamic redistribution and automated replication
- Security
  - Multiple sites, competing projects
  - GSS(GSI) - GGF security standards

Design of Grid Data Farm (4) --- System Components

- Gfarm Client – GridRPC and other client API
  - User/Administrator client program
  - GridRPC Client Toolkit / Gfarm Shell / Gfarm GUI
- Gfarm Server – Network Enabled Service and Global Data Management
  - Authentication / Scheduling / Program Execution
  - Management of gfarm pool and metadata
- Gfarm Pool – Gfarm (Parallel) Filesystem
  - Large Storage Cluster w/Gfarm Filesystem Daemon
- Gfarm MetaDB – Data Management Substrate
  - Gfarm Filesystem Metadata maintenance
  - Employ off-the-shelf RDB, OODB and LDAP/MDS

Gfarm Server (gfarmd)

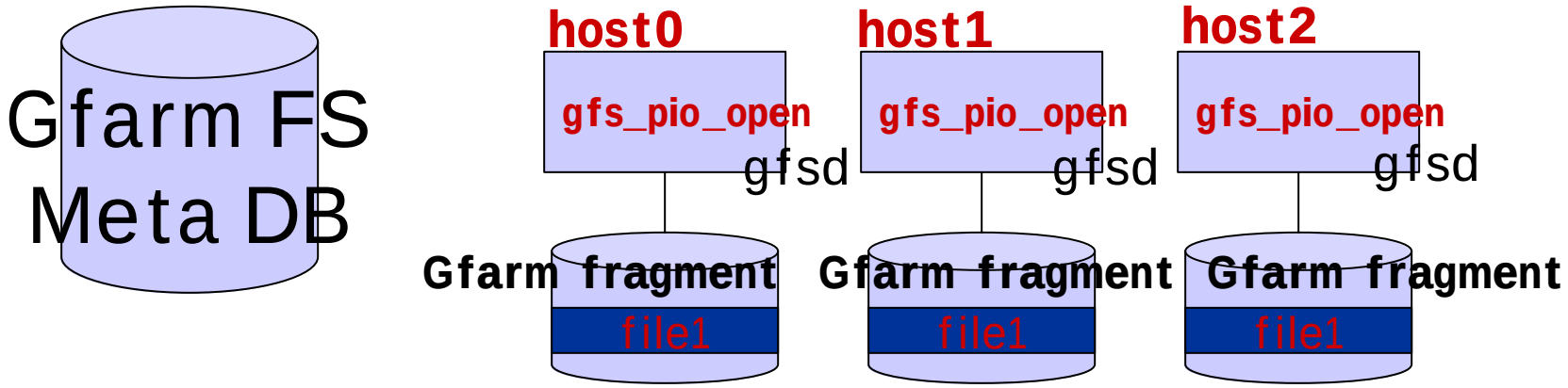
- Client Authentication / Authorization
  - GSI, GSS, SSH (and IPsec)
- Schedule machines using Gfarm FS Meta DB
- Invoke and control parallel programs efficiently



Gfarm Filesystem (Meta DB & gfsd)

- Gfarm URL: gfarm:~user/path/name
- Gfarm Parallel I/O API
- Gfarm FS Meta DB – Distributed File Management

gfs_pio_open_local("gfarm:/file1", "r");



Gfarm Parallel I/O APIs

- gfs_pio_create / open / close
- gfs_pio_create_local / open_local
- gfs_pio_read / write / seek / flush
- gfs_pio_getc / ungetc / putc
- gfs_mkdir / rmdir / unlink
- gfs_chdir / chown / chgrp / chmod
- gfs_stat
- gfs_opendir / readdir / closedir

Gfarm Security
- Single sign-on using GSSAPI

- gfarm-signon <gfarm_server>
  - Acquires a user credential using a Globus user certificate
    - Grid-proxy-init
  - Establishes a secure channel to the Gfarm server
  - Delegates the user credential to establish secure channels to Gfarm pool nodes
    - Currently, a passphrase is passed because GSSAPI does not have the APIs for credential delegation

Gfarm Commands (1)

- gfimport / gfexport
  - Copy a file to / from the Gfarm filesystem
- gfsched / gfwhere
  - List hostnames where each Gfarm fragment and replica is stored
- gfreg
  - Register a program or a file
- gfrep
  - Replicate a Gfarm file using Parallel I/O
- gfdigest
  - Check identity between master and replica

Gfarm Commands (2)

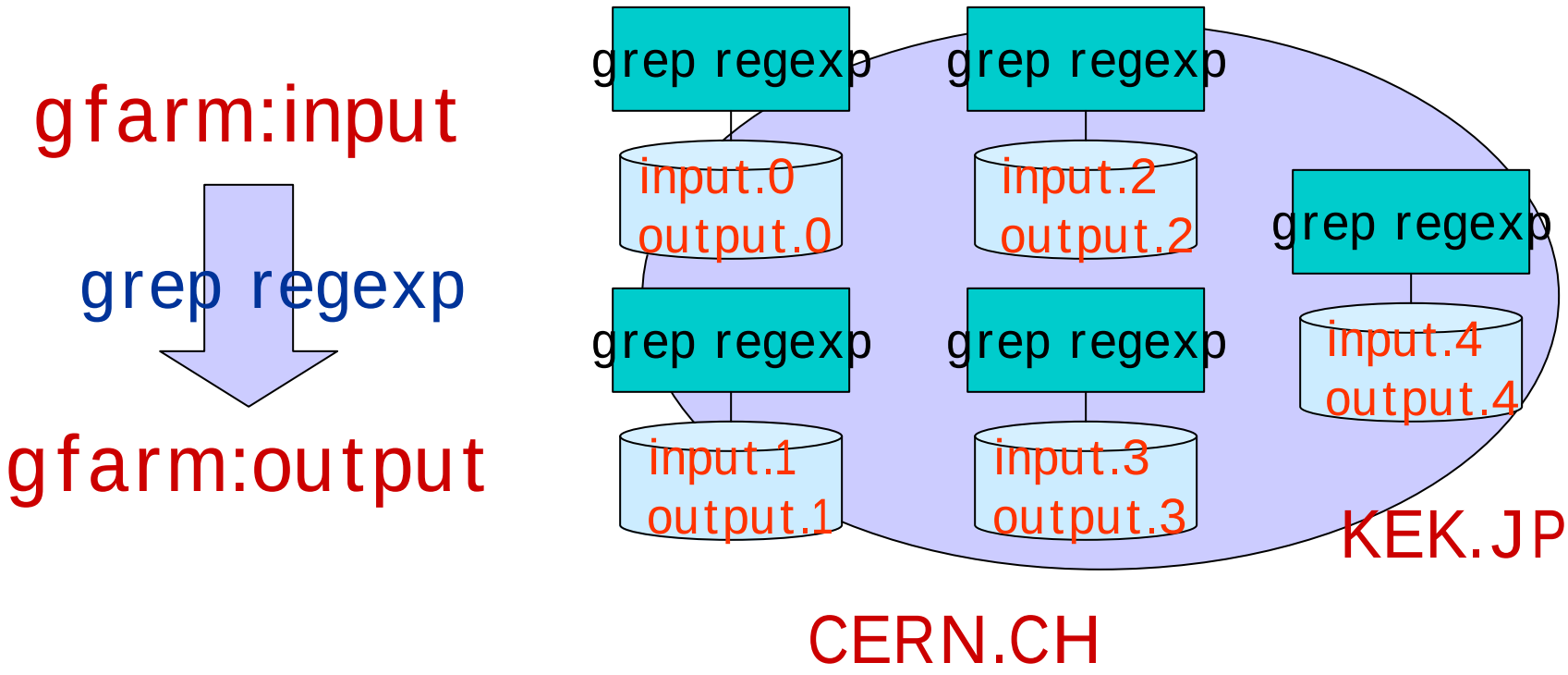
- gfls
  - Lists contents of directory
- gfrm, gfrmdir
  - Remove directory entries
- gfmkdir
  - Make directories
- gfcp
  - Copy files
- gfdf
  - Displays number of free disk blocks and files
- gfsck
  - Check and repair file systems

Gfarm Commands (3)

- gfgrep
  - Print lines matching a pattern
- gfwc
  - Print the number of bytes, words, and lines in files
  - Reduction in MPI

gfgrep --- parallel grep

% gfgrep –o gfarm:output regexp gfarm:input



Porting Legacy or Commercial Applications

- Hook syscalls open(), close(), … to utilize Gfarm filesystem
  - This allows thousands of files to be grouped automatically and processed in parallel.
  - Quick upstart for legacy apps but some portability problems have to be coped with
- gfreg command
  - After creation of thousands of files, gfreg explicitly groups files into a single Gfarm URL.

Gfarm Development Status

- Prototype v1 just completed, in Alpha Stage
  - Basic functionality of Gfarm filesystem has been implemented.
  - Can run test mock data challenge problems with **Objectivity** OODB!
  - Can be distributed to interested research groups
  - Is not entirely exclusive for LHC
- Several key features not implemented, or have rudimentary implementation
  - Better usage of Globus
- Collaborate with EU DataGrid and North American PPDG/GriPhyN/etc. efforts
  - Share development load, mutual utilization

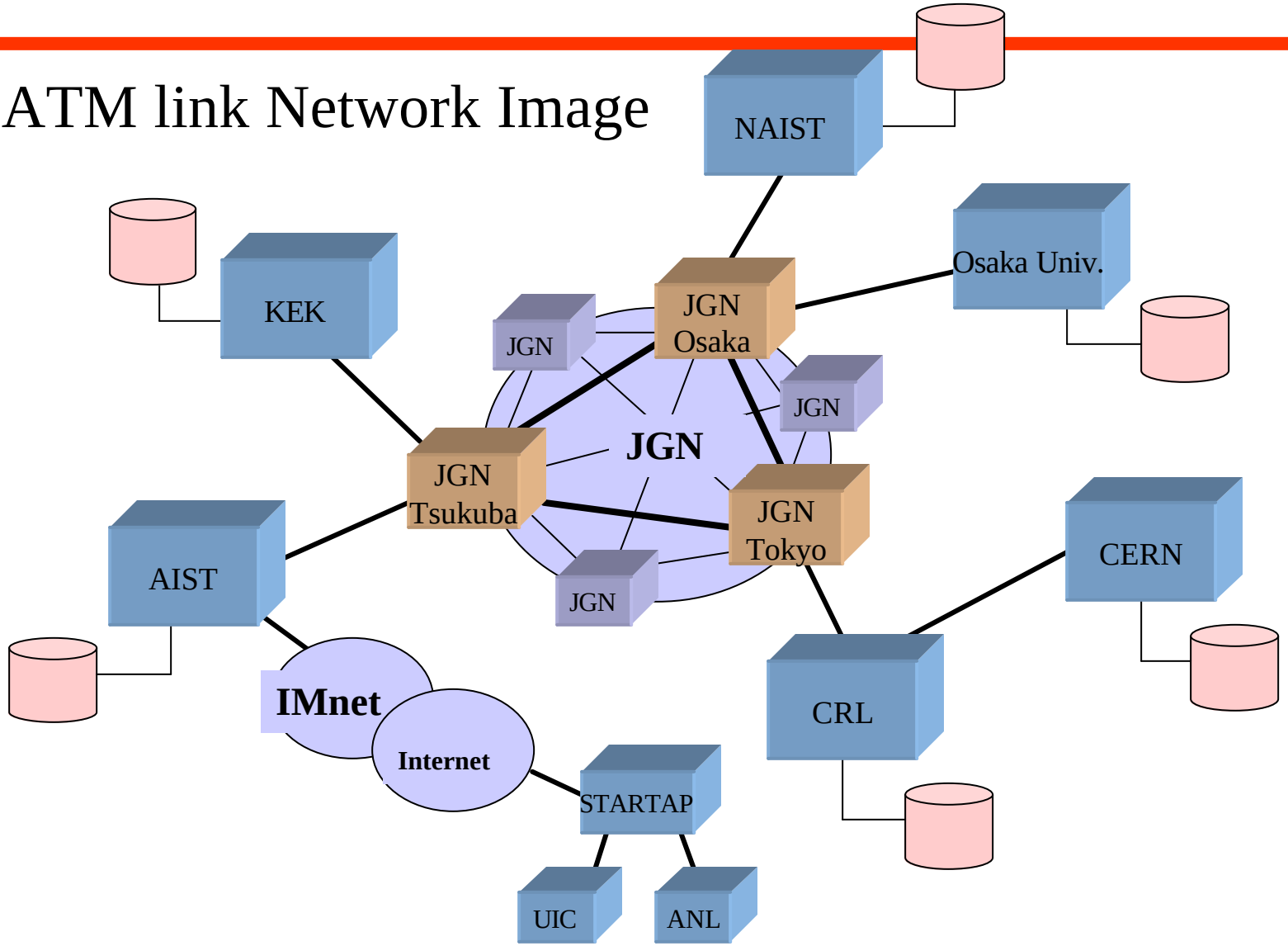
Gfarm target platform design and Grid testbed

Execution Platform: the Gfarm Cluster (Data Intensive Grid Cluster)

- Requirement: Petabytes of online storage
  - 2004-5 high-end PC technology, <$10 million
  - Installation at U.Tokyo to serve ATLAS Tier-1 Regional Center
  - Multiple Gfarm Node instantiations at different centers for other branches of science
    - Astronomy (Subaru Telescope, etc.), Earth Science, Genome Informatics, etc.
  - Must be cheap (< $10mil) -> Commodity Clustering Tech
- Linked by Tsukuba WAN/SuperSINET/JGN
  - 10GB/s by 2002 and beyond
  - Links dozens of Univ. and National Labs
  - Inter Gfarm coordination

Gfram: Grid Testbed

JGN ATM link Network Image



Initial Design of the Production Gfarm Cluster in 2005

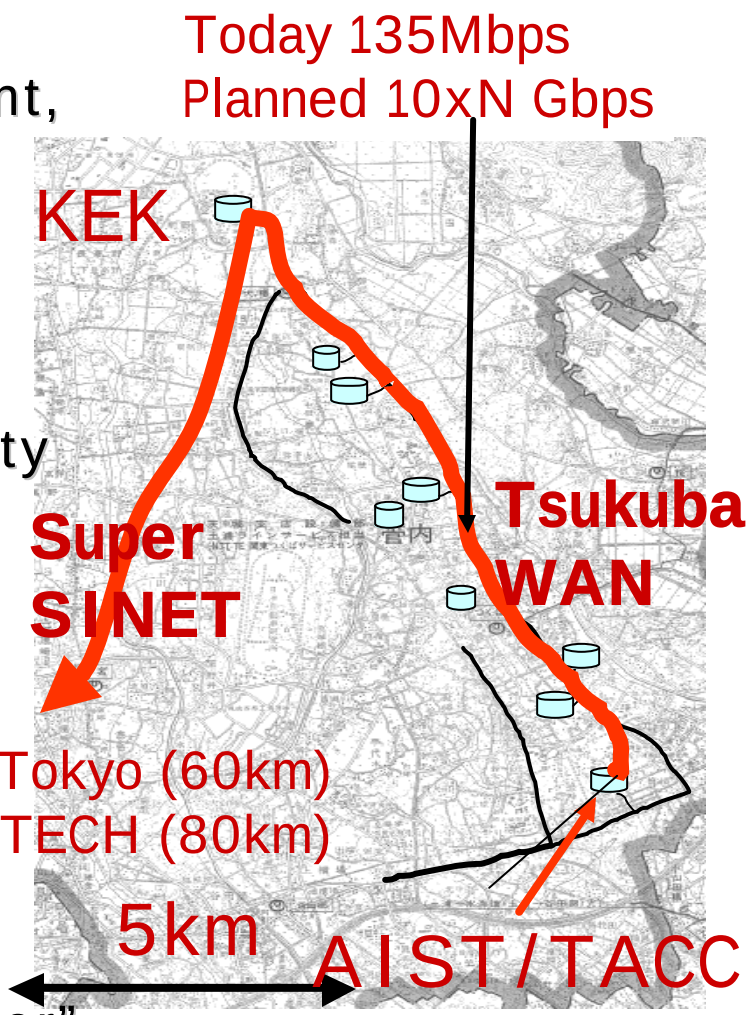
- Assume Infiniband or similar technology
- In-box CPUs and Disks locally interconnected via Infiniband
- Inter-box connection via Infiniband, direct interconnect into local fabric
- High-Density disk packaging required, cooling a big problem, need engineering development
- Somewhat different from business web server technology due to high computing and I/O capacity requirement

Initial Design of the Production Gfarm Cluster (2)

- Design of Gfarm Node
  - Commodity Technology circa 2005
  - 300GByte low power HD Drive, Raid 5, 25 Drives/box => 6 Terabytes/box (Plug&Play, Active Cooling)
  - >10GigaFlop SMT 64-bit CPUx4-8, >20GB RAM
  - Multi-channel, Multi-gigabit LAN, >10GigaBps
  - 4U box, 600W power/box, Active cooling
- Design of Gfarm System (2004-5 production)
  - 60TeraBytes@250 disks, 40CPUs/40U chassis, 5KWatts
  - 20 Chassis, 1.2 PetaBytes@5000HDDs, 8-16Teraflops @800CPUs, 100KWatts, 3 Petabyte Tape Storage
  - Direct Infiniband link into the WAN fabric

Grid Data Farm Development Schedule

- Initial Prototype 2000-2001
  - Gfarm filesystem, Metadata management, data streaming and GridRPC
  - Mock Data Challenge (Monte-carlo)
  - Deploy on Development Gfarm Cluster
- Second Prototype 2002(-2003)
  - Load balance, Fault Tolerance, Scalability
  - Accelerate by National "Broadband Computing initiative" proposal
- Full Production Development (2004-2005 and beyond)
  - Deploy on Production GFarm cluster
  - Petascale online storage
- Synchronize with ATLAS schedule
  - ATLAS-Japan Tier-1 RC "prime customer"



Summary

http://datafarm.apgrid.org/
datafarm@apgrid.org

- Petabyte-scale Data Intensive Computing wave of computational science
- Poses extreme challenges to HPC, current HPC solutions not well applicable
- Grid and Commodity Cluster technology as viable solutions
- Existing Grid infrastructure can be utilized, but further research and development required
- Grid Data Farm builds on success of Ninf (and other Grid projs. such as Globus) to cope with such challenge